Planar L-Drawings of Directed Graphs

Steven Chaplick, Markus Chimani, Sabine Cornelsen, Giordano Da Lozzo, Martin Nöllenburg, Maurizio Patrignani, Ioannis G. Tollis, Alexander Wolff

Graph Drawing and Network Visualization 2017 · Boston
Drawing Directed Graphs

There is a variety of drawing styles for directed graphs, e.g.

- **Layered layout**
  - [Sugiyama, Tagawa, Toda 1981]

- **Kandinsky style layout**
  - [Fößmeier, Kaufmann 1996]

- **Overloaded orthogonal layout**
  - [Kornaropoulos, Tollis 2011]
Drawing Directed Graphs

There is a variety of drawing styles for directed graphs, e.g.

- Layered layout
  [Sugiyama, Tagawa, Toda 1981]

- Kandinsky style layout
  [Fößmeier, Kaufmann 1996]

- Overloaded orthogonal layout
  [Kornaropoulos, Tollis 2011]

In 2016 Angelini et al. introduced **L-drawings**:

- exclusive x- and y-coordinates per vertex
- outgoing edges attach vertically
- incoming edges attach horizontally
- small arcs indicate L-bends
- crossings and “confluent” overlaps allowed
- exist for any graph
- ink minimization is NP-hard
Drawing Directed Graphs

There is a variety of drawing styles for directed graphs, e.g.

Layered layout
[Sugiyama, Tagawa, Toda 1981]

Kandinsky style layout
[Fößmeier, Kaufmann 1996]

Overloaded orthogonal layout
[Kornaropoulos, Tollis 2011]

In 2016 Angelini et al. introduced **L-drawings**:

- exclusive x- and y-coordinates per vertex
- outgoing edges attach vertically
- incoming edges attach horizontally
- small arcs indicate L-bends
- crossings and “confluent” overlaps allowed
- exist for any graph
- ink minimization is NP-hard
Planar L-Drawings

Definitions:

- **Planar L-drawing** if crossing-free
- **Upward planar L-drawing** if all edges y-increasing
- **Upward-rightward planar L-drawing** if all edges x- and y-increasing
Planar L-Drawings

Definitions:

- **Planar L-drawing** if crossing-free
- **Upward planar L-drawing** if all edges y-increasing
- **Upward-rightward planar L-drawing** if all edges x- and y-increasing

Observation:

Planar L-drawings correspond to planar 1-bend Kandinsky drawings with extra constraints on cyclic edge orders of vertices.
## Overview of Results

<table>
<thead>
<tr>
<th>Directed Planar Graphs</th>
<th>Planar $st$-Graphs</th>
<th>Directed Plane Graphs + Port Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar</td>
<td>NP-complete</td>
<td>Linear Time</td>
</tr>
<tr>
<td>Upward (-rightward) Planar</td>
<td>Characterization</td>
<td>Construction Linear Time Algorithm</td>
</tr>
</tbody>
</table>

- **Directed Planar Graphs**
  - Planar
  - Upward (-rightward) planar

- **Planar $st$-Graphs**
  - NP-complete

- **Directed Plane Graphs + Port Assignment**
  - Linear time
## Overview of Results

<table>
<thead>
<tr>
<th>Directed Planar Graphs</th>
<th>NP-complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar st-graphs</td>
<td>Characterization constructive linear time algorithm</td>
</tr>
<tr>
<td>Directed Plane Graphs + Port Assignment</td>
<td>Linear Time</td>
</tr>
</tbody>
</table>
Planar L-Drawings of Directed Graphs

Any planar L-drawing implies 4-modal embedding.

There are planar directed graphs that do not admit planar L-drawings.

6-modal in any embedding
Planar L-Drawings of Directed Graphs

Any planar L-drawing implies 4-modal embedding.

- There are planar directed graphs that do not admit planar L-drawings.

- There are graphs with 4-modal embedding but no planar L-drawing.

- every vertex is 4-modal . . .

- . . . but rightmost vertex in L-drawing can be at most bimodal

octahedron
NP-Completeness

**Theorem:** Deciding whether a directed graph admits a planar L-drawing is NP-complete.

**Proof:** (sketch)

- reduction from NP-complete **HV-rectilinear planarity testing**

[Didimo, Liotta, Patrignani 2014]

Given biconnected degree-4 planar graph $G$ with edges labeled $H$ and $V$, decide if $G$ admits drawing with horizontal $H$-edges and vertical $V$-edges.
NP-Completeness

**Theorem:** Deciding whether a directed graph admits a planar L-drawing is NP-complete.

**Proof:** (sketch)
- reduction from NP-complete **HV-rectilinear planarity testing** [Didimo, Liotta, Patrignani 2014]

Given biconnected degree-4 planar graph $G$ with edges labeled $H$ and $V$, decide if $G$ admits drawing with horizontal $H$-edges and vertical $V$-edges.

- core gadget: 4-wheel graph has basically two planar L-embeddings
NP-Completeness

**Theorem:** Deciding whether a directed graph admits a planar L-drawing is NP-complete.

**Proof:** (sketch)

- reduction from NP-complete *HV-rectilinear planarity testing* 
  
  Given biconnected degree-4 planar graph $G$ with edges labeled $H$ and $V$, decide if $G$ admits drawing with horizontal $H$-edges and vertical $V$-edges.

- core gadget: 4-wheel graph has basically two planar L-embeddings
- in $HV$-graph $G$ replace vertices by 4-wheel and edges by $H$-/V-gadgets
**NP-Completeness**

**Theorem:** Deciding whether a directed graph admits a planar L-drawing is NP-complete.

**Proof:** (sketch)
- reduction from NP-complete **HV-rectilinear planarity testing**

[Didimo, Liotta, Patrignani 2014]

Given biconnected degree-4 planar graph $G$ with edges labeled $H$ and $V$, decide if $G$ admits drawing with horizontal $H$-edges and vertical $V$-edges.

- core gadget: 4-wheel graph has basically two planar L-embeddings
- in $HV$-graph $G$ replace vertices by 4-wheel and edges by $H$/-$V$-gadgets

resulting graph $G'$ has planar L-drawing $\iff$ $G$ has $HV$-drawing
## Overview of Results

<table>
<thead>
<tr>
<th>Directed Planar Graphs</th>
<th>Planar st-graphs</th>
<th>Directed Plane Graphs + Port Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar</td>
<td>NP-complete</td>
<td>Linear Time</td>
</tr>
<tr>
<td>Upward (-rightward) planar</td>
<td></td>
<td>Characterization constructive linear time algorithm</td>
</tr>
</tbody>
</table>
Planar \( st \)-Graphs and Bitonic \( st \)-Orderings

- A **planar \( st \)-graph** \( G \) is a directed acyclic graph with exactly one source \( s \) and one sink \( t \), both embeddable on same face.
Planar $st$-Graphs and Bitonic $st$-Orderings

- A **planar $st$-graph** $G$ is a directed acyclic graph with exactly one source $s$ and one sink $t$, both embeddable on same face.

- **Planar $st$-graphs** always admit **straight-line upward planar drawings**

  \[ \text{[Di Battista, Tamassia 1988]} \]

- **Have $st$-ordering** $\pi$ respecting edge directions
Planar \( st \)-Graphs and Bitonic \( st \)-Orderings

- A **planar \( st \)-graph** \( G \) is a directed acyclic graph with exactly one source \( s \) and one sink \( t \), both embeddable on same face.

- **Planar \( st \)-graphs always admit straight-line upward planar drawings**

- **Have \( st \)-ordering \( \pi \) respecting edge directions**

- **\( st \)-ordering \( \pi \) of plane \( st \)-graph (planar \( st \)-graph + embedding) is bitonic** if successors of each vertex form bitonic sequence


\[ [\text{Di Battista, Tamassia 1988}] \]

\[ [\text{Gronemann 2014, 2016}] \]
Planar $st$-Graphs and Bitonic $st$-Orderings

- A **planar $st$-graph** $G$ is a directed acyclic graph with exactly one source $s$ and one sink $t$, both embeddable on same face.

- Planar $st$-graphs always admit straight-line upward planar drawings [Di Battista, Tamassia 1988]

- Have **$st$-ordering** $\pi$ respecting edge directions

- $st$-ordering $\pi$ of plane $st$-graph (planar $st$-graph + embedding) is **bitonic** if successors of each vertex form bitonic sequence [Gronemann 2014, 2016]

- For a planar $st$-graph $G$ define a **bitonic pair** $(\mathcal{E}, \pi)$ as an upward planar embedding $\mathcal{E}$ of $G$ with a bitonic $st$-ordering $\pi$. 

![Diagram](image-url)
Characterization

**Theorem:** A planar $st$-graph admits an upward-planar L-drawing if and only if it admits a bitonic pair.
Characterization

**Theorem:** A planar $st$-graph admits an upward-planar L-drawing if and only if it admits a bitonic pair.

**Proof:** (sketch)
Characterization

Theorem: A planar \( st \)-graph admits an upward-planar L-drawing if and only if it admits a bitonic pair.

Proof: (sketch)

\[ \Rightarrow \]

- y-coordinates induce \( st \)-ordering \( \pi \)
- \( \pi \) is bitonic due to upward L-properties
Characterization

**Theorem:** A planar \(st\)-graph admits an upward-planar L-drawing if and only if it admits a bitonic pair.

**Proof:** (sketch)

\[ \Rightarrow \]

- y-coordinates induce \(st\)-ordering \(\pi\)
- \(\pi\) is bitonic due to upward L-properties

\[ \Leftarrow \]

- use \(\pi\) for y-coordinates
- incrementally construct partial order \(\prec\) as basis for x-coordinates

\[ G_{i-1} \]

\[ G_i \]

- invariant: outer face of \(G_i\) simple cycle ordered by \(\prec\)
- insert each \(v_i\) btw. last two predecessors
Characterization

**Theorem:** A planar \( st \)-graph admits an upward-planar L-drawing if and only if it admits a bitonic pair.

**Proof:** (sketch)

\[ \Rightarrow \]

- y-coordinates induce \( st \)-ordering \( \pi \)
- \( \pi \) is bitonic due to upward L-properties

\[ \Leftarrow \]

- use \( \pi \) for y-coordinates
- incrementally construct partial order \( \prec \) as basis for x-coordinates

invariant: outer face of \( G_i \) simple cycle ordered by \( \prec \)

special case: just one predecessor \( \rightarrow \) augment graph similar to [Gronemann 2016]
Finding Bitonic Pairs

- **Assumption:** st-graph $G$ is biconnected and has edge $(s, t)$
- We say $G$ is $v$-monotonic or (strictly) $v$-bitonic if for every vertex $v$, subgraph induced by successors of $v$ (− transitive edges) is a path $p$
- $p$ is monotonic or (strictly) bitonic

- **Diagram:**
  - $v$
  - Apex of $v$ and $p$

- **Inclusion:**
  - Monotonic $\subseteq$ Bitonic
  - Strictly Bitonic $\subseteq$ Bitonic
Finding Bitonic Pairs

- **Assumption:** \( st \)-graph \( G \) is biconnected and has edge \((s, t)\)

- We say \( G \) is \( v \)-monotonic or (strictly) \( v \)-bitonic if for every vertex \( v \)
  - subgraph induced by successors of \( v \) (− transitive edges) is a path \( p \)
  - \( p \) is monotonic or (strictly) bitonic

\[
\text{monotonic} \subset \text{bitonic} \quad \text{strictly bitonic} \subset \text{bitonic}
\]

- **Goal:** augment \( G \) into \( G^* \) by adding edges s.t. \( G^* \) is \( v \)-bitonic
Finding Bitonic Pairs

- **Assumption:** \( st \)-graph \( G \) is biconnected and has edge \((s, t)\)

- We say \( G \) is \( v \)-**monotonic** or (strictly) \( v \)-**bitonic** if for every vertex \( v \)
  - subgraph induced by successors of \( v \) (− transitive edges) is a path \( p \)
  - \( p \) is **monotonic** or (strictly) **bitonic**

- **Goal:** augment \( G \) into \( G^* \) by adding edges s.t. \( G^* \) is \( v \)-bitonic

### Theorem:
Plane \( st \)-graph \( G \) admits bitonic \( st \)-ordering iff \( G^* \) is \( v \)-bitonic.
Any \( st \)-ordering of \( G^* \) is bitonic \( st \)-ordering of \( G \).  

\( \sim \) [Gronemann 2016]

\( \rightarrow \) The task of finding a bitonic pair of \( G \) reduces to finding an augmentation \( G^* \) of \( G \) that is \( v \)-bitonic.
Finding a $\nu$-Bitonic Augmentation

Visiting the SPQR-tree of biconnected planar $st$-graph $G$ rooted at edge $(s, t)$ in bottom-up fashion find augmentation $G^*$ and embedding (if one exists).
Finding a \( v \)-Bitonic Augmentation

Visiting the SPQR-tree of biconnected planar \( st \)-graph \( G \) rooted at edge \( (s, t) \) in bottom-up fashion find augmentation \( G^* \) and embedding (if one exists).

An SPQR-node \( \mu \) with source \( s_\mu \) is of

- **type M** if the augmented pertinent graph is \( s_\mu \)-monotonic
- **type B** if the augmented pertinent graph is strictly \( s_\mu \)-bitonic
Finding a \( v \)-Bitonic Augmentation

Visiting the SPQR-tree of biconnected planar \( st \)-graph \( G \) rooted at edge \((s, t)\) in bottom-up fashion find augmentation \( G^* \) and embedding (if one exists).

An SPQR-node \( \mu \) with source \( s_\mu \) is of

- **type M** if the augmented pertinent graph is \( s_\mu \)-monotonic
- **type B** if the augmented pertinent graph is strictly \( s_\mu \)-bitonic

When processing an SPQR-node \( \mu \) our primary goal is to make it type M and otherwise type B. If both fails, no \( v \)-bitonic augmentation of \( G \) exists.
Processing SPQR-Nodes

Q-node: trivially type M
Processing SPQR-Nodes

**Q-node:** trivially type M

**S-node:** replace each virtual edge by augmented pertinent graph of child node with arbitrarily flipped embedding. Node type is inherited from bottom child.
Processing SPQR-Nodes

**Q-node:** trivially type M

**S-node:** replace each virtual edge by augmented pertinent graph of child node with arbitrarily flipped embedding. Node type is inherited from bottom child.

**P-node:**
- if two or more children are of type B
  → successors of $s_\mu$ have two apices, so regardless of embedding no $s_\mu$-bitonic augmentation exists
Processing SPQR-Nodes

**Q-node:** trivially type M

**S-node:** replace each virtual edge by augmented pertinent graph of child node with arbitrarily flipped embedding. Node type is inherited from bottom child.

**P-node:**
- if two or more children are of type B
  \[\rightarrow \text{successors of } s_{\mu} \text{ have two apices, so regardless of embedding no } s_{\mu}\text{-bitonic augmentation exists}\]
- if one child is of type B and one child is Q-node for \((s_{\mu}, t_{\mu})\)
  \[\rightarrow \text{apex of type-B node } \neq t_{\mu}, \text{ but } t_{\mu} \text{ must be apex of } s_{\mu};\]
  again \(s_{\mu}\) has two apices and no \(s_{\mu}\)-bitonic augmentation exists
Processing SPQR-Nodes

**Q-node:** trivially type M

**S-node:** replace each virtual edge by augmented pertinent graph of child node with arbitrarily flipped embedding. Node type is inherited from bottom child.

**P-node:**

- else embed child of type B or Q-node for \((s_\mu, t_\mu)\) rightmost (if any) and connect successors of \(s_\mu\) in order of embedding; node type is M \(\iff\) rightmost child is of type M
Processing SPQR-Nodes

**Q-node:** trivially type M

**S-node:** replace each virtual edge by augmented pertinent graph of child node with arbitrarily flipped embedding. Node type is inherited from bottom child.

**P-node:**
- if two or more children are of type B
  - successors of $s_\mu$ have two apices, so regardless of embedding no $s_\mu$-bitonic augmentation exists
- if one child is of type B and one child is Q-node for $(s_\mu, t_\mu)$
  - apex of type-B node $\neq t_\mu$, but $t_\mu$ must be apex of $s_\mu$; again $s_\mu$ has two apices and no $s_\mu$-bitonic augmentation exists
- else embed child of type B or Q-node for $(s_\mu, t_\mu)$ rightmost (if any) and connect successors of $s_\mu$ in order of embedding; node type is M $\iff$ rightmost child is of type M

**R-node:** more complicated, see paper
Upward Planar L-Drawings

**Theorem:** It can be tested in linear time whether a planar $st$-graph $G$ admits an upward-planar L-drawing. If it does, it can also be constructed in linear time.

**Proof:** (sketch)
- process SPQR-tree to find $v$-bitonic augmentation $G^*$ and embedding $E^*$ in root node (if any)
- any $st$-ordering $\pi$ of $G^*$ yields bitonic pair $(E, \pi)$ of $G$
- $G$ has bitonic pair $\iff G$ admits upward-planar L-drawing
- all steps can be implemented in linear time
Upward Planar L-Drawings

**Theorem:** It can be tested in linear time whether a planar $st$-graph $G$ admits an upward-planar L-drawing. If it does, it can also be constructed in linear time.

**Proof:** (sketch)
- process SPQR-tree to find $v$-bitonic augmentation $G^*$ and embedding $E^*$ in root node (if any)
- any $st$-ordering $\pi$ of $G^*$ yields bitonic pair $(E, \pi)$ of $G$
- $G$ has bitonic pair $\iff G$ admits upward-planar L-drawing
- all steps can be implemented in linear time

**Remark:** Same approach can be used to decide existence and construct upward-rightward-planar L-drawings.
## Summary

<table>
<thead>
<tr>
<th>Directed planar graphs</th>
<th>NP-complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar $st$-graphs</td>
<td>Characterization, constructive linear time algorithm</td>
</tr>
<tr>
<td>Directed plane graphs</td>
<td>Linear time → see paper</td>
</tr>
<tr>
<td>+ Port assignment</td>
<td></td>
</tr>
<tr>
<td>Directed plane graphs</td>
<td></td>
</tr>
<tr>
<td>Upward planar graphs</td>
<td></td>
</tr>
<tr>
<td>Bimodal graphs</td>
<td></td>
</tr>
</tbody>
</table>

Notation:
- $st$-graphs: Directed planar graphs with a specified source and target.
- NP-complete: A complexity class for problems that can be solved in polynomial time if a proposed solution is given, otherwise the problem is computationally hard.
- Characterization: A formal description or classification of the objects or phenomena studied.
- Constructive linear time algorithm: An algorithm that constructs a solution in linear time relative to the input size.

Diagram:
- Planar: A graph that can be drawn on a plane without any edges crossing.
- Upward (rightward) planar: A planar graph where the edges are directed from left to right.

[Chaplick, Chimani, Cornelsen, Da Lozzo, Nöllenburg, Patrignani, Tollis, Wolff - Planar L-Drawings of Directed Graphs]
<table>
<thead>
<tr>
<th>Summary</th>
<th>planar</th>
<th>upward (-rightward) planar</th>
</tr>
</thead>
<tbody>
<tr>
<td>directed planar graphs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>planar $st$-graphs</td>
<td>NP-complete</td>
<td></td>
</tr>
<tr>
<td>directed plane graphs + port assignment</td>
<td>linear time</td>
<td>\rightarrow see paper</td>
</tr>
<tr>
<td>directed plane graphs</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>upward planar graphs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bimodal graphs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>